

Practical 4: Unstructured combinational circuits

4.1 About this practical

The objective of this practical is to reinforce your knowledge of standard forms of logic functions. We will study SoP, PoS, NAND and NOR implementations of combinational circuit. We will use a 7-segment display driver circuit as an example.

Contents

4.1	About this practical	1
4.2	A 7-segment display driver circuit	1
4.3	The function table and Karnaugh maps	2
4.4	Creating a block diagram	3
4.5	Simulating the 7-segment driver	4
4.6	Generate the VHDL code for the 7-segment driver	4
4.7	The report	4

4.2 A 7-segment display driver circuit

(You can read problem 4-9 from the textbook for an additional background material. Also sections 4-2 and 4-3 can be helpful.)

- The **seven-segment display** is a common display device for decimal numbers that you can find in many devices like digital watches, etc.
- The seven-segment display **driver circuit** converts a binary-coded-decimal (BCD) digit, $x(3 : 0)$, into seven binary signals (a, b, c, d, e, f, g) that drive (highlight) respective segments of the display, as shown in Figure 1.

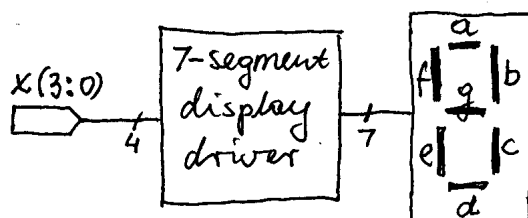


Figure 1: A 7-segment display with the driver

4.3 The function table and Karnaugh maps

- The next step is to create the function/truth table for all seven output signals. We start with the representation of decimal digit by the combination of segments as in Figure 2.



Figure 2: 7-segment display representation of decimal digits

From the above representation we can create the following table:

x	x_3	x_2	x_1	x_0	selected seg's	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	0	0	0	<i>a, b, c, d, e, f, g</i>	1	1	1	1	1	1	0
1	0	0	0	1	<i>b, c</i>	0	1	1	0	0	0	0
2	0	0	1	0	<i>a, b, d, e, g</i>	1	1	0	1	1	0	1
3	0	0	1	1	<i>a, b, c, d, g</i>	1	1	1	1	0	0	1
4	0	1	0	0	<i>b, c, f, g</i>	0	1	1	0	0	1	1
5	0	1	0	1	<i>a, c, d, f, g</i>	1	0	1	1	0	1	1
6	0	1	1	0	<i>a, c, d, e, f, g</i>	1	0	1	1	1	1	1
7	0	1	1	1	<i>a, b, c</i>	1	1	1	0	0	0	0
8	1	0	0	0	<i>a, b, c, d, e, f, g</i>	1	1	1	1	1	1	1
9	1	0	0	1	<i>a, b, c, d, f, g</i>	1	1	1	1	0	1	1
10	1	0	1	0	none	0	0	0	0	0	0	0
11	1	0	1	1	none	0	0	0	0	0	0	0
12	1	1	0	0	none	0	0	0	0	0	0	0
13	1	1	0	1	none	0	0	0	0	0	0	0
14	1	1	1	0	none	0	0	0	0	0	0	0
15	1	1	1	1	none	0	0	0	0	0	0	0

- The above table can be now converted into seven 4-input Karnaugh maps.
- In order to have the smaller design assume that input combinations 10, . . . , 15 are “**don’t care**” conditions that can take any value, 0, or 1 to help to minimise the function.
- From the Karnaugh maps derive simplified equations in
 - SoP form, and
 - PoS form.

Include these two forms for **all seven functions in your report**.

- From the above forms derive two equivalent implementations
 - NAND only, and
 - NOR only implementation.

4.4 Creating a block diagram

- To create the logic diagrams for all seven functions in two forms is impossible to complete in a sensible time. Therefore you will need to implement only **three selected functions**.
- The choice of functions to implement is based on the **last digit of your student number d** .
- Calculate

$$s_1 = d \bmod 7, \quad s_2 = (d + 3) \bmod 7, \quad s_3 = (d + 5) \bmod 7$$

- Select functions to implement from the following table

s	0	1	2	3	4	5	6
$7s$	a	b	c	d	e	f	g

- Select the NAND implementation if d is **even**.
Select the NOR implementation if d is **odd**.
- Follow the methodology from the previous pracs and build the block/logic diagram for the three selected functions.
- Try to arrange your design in a way similar to that in Figure 3.

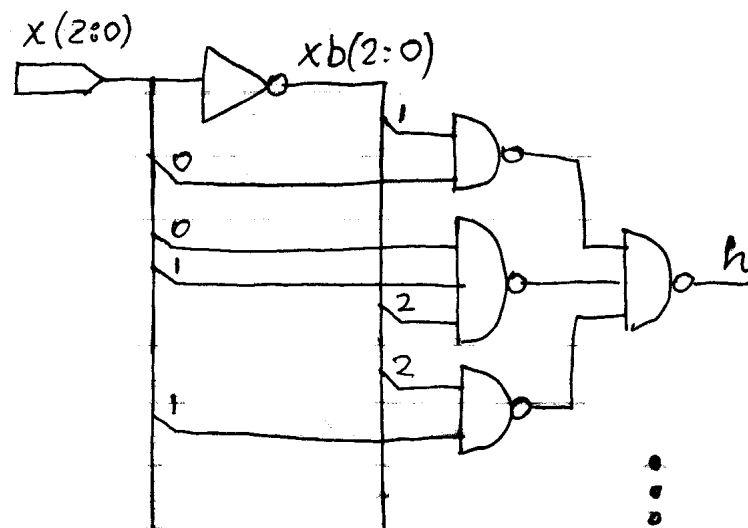


Figure 3: 7-segment display representation of decimal digits

4.5 Simulating the 7-segment driver

Write a simulation script and produce the time waveform that demonstrate the correctness of your design.

4.6 Generate the VHDL code for the 7-segment driver

Generate the VHDL code for your implementation and inspect it.

4.7 The report

In your report (due after prac 4) include the results in the form of:

- Logic equations as specified in this manual,
- block/logic diagrams,
- VHDL programs (if available),
- simulation scripts (if available),
- simulation waveforms,
- **short** description of the above.

Wherever possible publish the results selecting in the **Block Diagram** window

File → **HTML Export ...**. Specify the export target directory to be `... \DigDes \Reports`.